Proposal to the Department of Energy (DOE) Office of Science
Notice LAB 99-08: Next Generation Internet
Research In Basic Technologies

Advanced Visualization Communication Toolkit

Lawrence Berkeley National Laboratory
Berkeley, CA 94720

Year 1: $284,918
Year 2: $295,319 (if funding available)
Year 3: $306,143 (if funding available)
Total Request: $886,380 (if funding available)

Principal Investigators:

Deborah Agarwal
Information and Computing Sciences Division
1 Cyclotron Rd, MS 50B-2239
Berkeley, CA 94720
Phone: +1 (510) 486-7078
Fax: +1 (510) 486-6363
Email: DAAgarwal@lbl.gov

_____
3/29/99

Stephen Lau, Jr.
NERSC Division
1 Cyclotron Rd, MS 50F
Berkeley, CA 94720
Phone: +1 (510) 486-7178
Fax: +1 (510) 486-5548
Email: SLau@lbl.gov

_____
3/29/99

Official:

Stewart Loken
Director of Information and Computing Sciences Division
Information and Computing Sciences Division
Phone: +1 (510) 486-7474
Fax: +1 (510) 486-4300
Email: SCLoken@lbl.gov

_____
3/29/99

Human subjects: NO
Use of vertebrate animals: NO

Table of Contents

Abstract

One of the goals of the next generation Internet is to provide seamless collaborative access to data and to visualize data remotely where the data must travel across a wide area network to reach the user. Such remote visualization requires the development of visualization software that adapts to the dynamics of the underlying networking infrastructure. The data visualization communication toolkit will allow visualization applications to have direct access to network status information and to directly control communication protocol and network behavior. Visualization applications will be able to adapt to a changing networking infrastructure and provide real-time feedback to the user regarding the state of the network. The toolkit will also provide options for the visualization software to modify its behavior to reflect the current networking conditions. These capabilities will also allow for there to be multiple sites with different network characteristics to be viewing the data simultaneously, allowing collaboration

1 Introduction

Today's interactive scientific visualizations are data intensive and are rendered at high resolutions at a rate of at least 30 frames per second. These visualizations require high bandwidth between the data source and the rendering engine and between the rendering engine and the viewer. With the ever increasing computation capabilities of today's supercomputers the size of the data set to be visualized is also increasing. The next generation Internet will provide the opportunity for the visualization data, rendering engine and viewer to be located at different sites. Not having these resources at the same site introduces network bandwidth, loss and latency problems that do not occur when the resources are at a single site. Remote visualization will demand new visualization software that is adaptable to changing network conditions and provides a closer coupling to the communication infrastructure.

The remote visualization environment which incorporates networked resources has been referred to as a data visualization corridor. The goal of data visualization corridors is to provide visualization capabilities that are well beyond what is available today. A principal goal is to develop hierarchical decompositions so that the visualization can adapt to the current network capabilities. These new visualizations will require not only improved network throughput but they will also require new communication protocols to support dynamic evaluation of and adaptation to the changing network environment. The use of existing protocols and a differentiated services capable network will not be enough to satisfy the requirements of these new visualizations particularly in collaborative situations where the users are likely to have different network capabilities and availabilities. If all users are required to have a high bandwidth connection and services like differentiated services available, then the ability to do collaborative visualizations will be severely limited. Also, even with all users having high bandwidth networks, the amount of bandwidth available to a particular application at a particular time is likely to vary. The visualization software needs to be able to determine the bandwidth available and adapt to the current capabilities.

The advanced visualization communication toolkit will provide the abstractions and the

underlying mechanisms needed to allow control of the reliability, duplicity, timing and priority of the visualization data. These capabilities will be accessed through a visualization specific interface that allows the application to operate in its own language of images and data rather than raw communication capabilities. It will enable existing and new visualization packages to effectively service remote users and to react to the dynamically changing network environment. As an example, the toolkit interface will allow the visualization software to specify the visualization context of the data so that appropriate priorities can be assigned to the data. For example, the toolkit will not waste resources retransmitting data from a frame that is no longer being viewed.

Using knowledge of the relative importance of the different portions of the data the toolkit will ensure that the most important data is sent with redundant encoding and is retransmit with the highest priority. Each level of a hierarchical visualization will be assigned an appropriate priority. The toolkit will also allow the visualization software to determine the capabilities of participating users so that sites with lower bandwidth connections or with lesser visualization functionality will receive only the data they can handle and will be able to participate without forcing all users in a shared visualization to operate at the functionality of the user with the lowest available bandwidth and resolution.

A key requirement in interactive systems is low latency. The inherent transmission times of wide area networks and the large size of the data or pre-rendered images means that the visualization system must carefully plan its transmissions to avoid latency problems. In an immersive system, variations in latency or large latencies can cause "cybersickness" where the viewer's visual reference system becomes disoriented and begins to feel ill. In a non-immersive system, latency problems can quickly make a system unusable.

By providing the visualization software with information regarding the network latency and available bandwidth, the visualization communication toolkit will allow the visualization system to adapt. In some cases, it may be more efficient for the system to transmit low resolution images across the network, in other instances; it may be more efficient to transmit a portion of the data set for rendering at the remote user's site. In the extreme case, the system would warn the user that the underlying network is not able to perform the desired operation. Even in this situation, the visualization system will at least be able to tell the user why they won't be able to complete an operation: current network based visualization systems do not provide this information.

The toolkit will be responsible for reserving available bandwidth when it is requested by the visualization software. As remote users join the session, the additional connections will be translated into multicasts with the appropriate reliability mechanisms. Reliable messaging normally produces higher latencies whenever a message is lost since the message must be retransmit before the following messages can be delivered. The visualization toolkit will make use of application level framing to identify messages with their visualization context. The toolkit will use the framing identifier of lost messages to determine whether they are still needed by the visualization.

The visualization toolkit will also provide priority mechanisms. This allows the

visualization system to identify the most important messages or the messages with the most restrictive latency constraints as high priority. The communication layer will handle these messages based on their priorities. A high priority message will be moved forward on the output stack and redundant encoding such as FEC will be used to improve the likelihood that all the data will arrive without requiring retransmission.

The communication library that is needed to support the described capabilities of the visualization toolkit is significantly different from what is available today. Most current communication protocols provide generic services and work very hard to hide the network dynamics from the user. Their goal is to provide a very simple message-passing interface. The application is given only rudimentary control over the flow control, priority and retransmission decisions. These simple interfaces are appropriate for most applications and the service provided is a reasonably good fit with their needs. The idea of the visualization toolkit is to instead provide a two part solution. A generic low-level communication protocol toolkit that provides accurate feedback regarding current network conditions, and fine grained control over the message handling, and a high-level visualization toolkit that uses the communication layer to provide generic easy to use visualization API that is appropriate for remote visualization.

Incorporating the visualization toolkit into existing visualization software will require a detailed analysis of the types of data that will be transmitted across the network. This analysis will help to determine how to construct the messages, how to prioritize the data for transmission and how to make the visualization applications network aware. Some of the types of visualization data that will need to be analyzed will be user interaction, geometry, raw data and rendered image data. There will be some instances where a hiearchical representation will be appropriate, and other instances where it will not. Dependent upon the type of data being visualized and the underlying networking infrastructure, there will need to be developed a mechanism that will be able to maximize the use of the information provided from the visualization toolkit.

2 Background and Significance

2.1 Communication Protocols

2.1.1 Unicast TCP and UDP

The most widely used protocols today are the traditional unicast protocols such as UDP and TCP which provide point-to-point delivery. The UDP protocol provides an unreliable datagram service and TCP provides a reliable stream. With a UDP connection, messages are sent with best effort delivery. There is no congestion control in the protocol to allow adaptation to the current network congestion level. A TCP connection provides for reliable ordered delivery of bits from sender to receiver. TCP makes the assumption that the connection is simply a stream and that there are no explicit message boundaries. The semantics of TCP completely rule out prioritizing messages within a stream and delivering messages out of order. TCP and UDP are only intended for use in connecting two processes. These protocols are inefficient when used to send messages to multiple

destinations. With TCP and UDP, a message intended for multiple destinations must be copied and a separate copy of the message sent to each individual receiver.

2.1.2 IP Multicast

IP multicast is a capability that has recently been added to the Internet. It provides a best effort datagram delivery service to a multicast group. The multicast messages are duplicated as needed in the routers to reach the members of the group. One difficulty with using IP multicast is that not all routers in the Internet implement the IP multicast routing protocol and many routers have an early version of the routing protocols that is inefficient. As IP multicast routing moves into more common use this problem will gradually go away and IP multicast routing will be relatively ubiquitous. In the meanwhile software that relies on IP multicast capabilities needs to provide a means of bridging in group members that can only receive unicast connections. This bridging capability is likely to also be needed to support users that are behind a firewall. When there are multiple sites that need to receive data by sending the data using IP multicast the load on the sending processor and the network bandwidth required can both be reduced. This savings becomes significant when there are large amounts of data as in a visualization.

2.1.3 Reliable Multicast

Message reliability and ordering semantics do not directly translate from TCP to the multicast environment. In a multicast group there can be multiple senders and the application may need messages to be delivered in order by source, delivered totally ordered within the group, or delivered unordered. Reliable delivery of messages also has many possible definitions in a multicast environment. The weakest reliable delivery protocol mechanisms simply provide a mechanism for the application to request retransmission of messages. These protocols do not provide any guarantees that the message was delivered to any of the group members or that the message is available for retransmission. The most well-known of these protocols is SRM [11]. In this protocol it is left up to the next higher layer to decide when to request retransmissions and when a message can be discarded. Application level framing (ALF) is used in conjunction with SRM to provide the application with a means of identifying the context of each message and determine whether a missing message needs to be retransmit.

The strongest reliable delivery mechanisms take care of retransmissions in the protocol layer so that the application is unaware of missing messages. These protocols include Totem[4][6], Transis[10], Xpress Transport Protocol (XTP)[21], Spread [7] and Reliable Multicast Protocol (RMP) [26]. Totem, Spread, and RMP also provide a delivery service that indicates when the message has been received at all of the group members . The various combinations of levels of ordered and reliable message delivery are suited to particular classes of applications. Since each protocol is usually built with a target application in mind, a wide variety of reliable multicast protocols have been built; each with slightly different service guarantees.

Reliable multicast protocols often also provide group membership maintenance and notification services. These services determine the current group membership for the application. The protocols that provide reliable delivery need to maintain some form of membership to determine when a message has been received by all of the group members. Most of these protocols pass this information to the application. The protocols that provide notification of when a message has been received at all the members of the group usually pass membership to the application as well since otherwise the guarantee is relatively meaningless. Protocols like SRM intentionally do not maintain membership to help make the protocol scalable. Other protocols like XTP maintain membership but do not provide notification of the membership to the application. A majority of the other reliable ordered delivery protocols maintain strict membership of the group and deliver notification of membership changes in order with respect to the data messages sent within the group.

Another class of reliable multicast protocols is designed to provide reliable file transfer. These protocols concentrate on efficient transfer of bulk data with no consideration for timing constraints. Two of the better known file transfer protocols are the Multicast File Transfer Protocol (MFTP) [19] and the Multicast Dissemination Protocol (MDP) [28]. These protocols send their data in rounds. They send each file as a bulk message and then go through rounds of retransmissions to ensure reliable delivery. Although these protocols do allow overlapping transmission of different files, they are not intended to provide a real-time service. In these protocols, the ordering of packets is only in the context of a particular file.

Prioritization of messages is not meaningful in the context of totally ordered messages since the protocol would be violating its delivery constraints if it delivered the messages out of order. The XTP protocol does allow the user to specify priority on messages and since XTP has both unicast and multicast communication protocols, the priority settings allow the protocol to determine and act based on the relative importance of the connections it is handling.

2.1.4 Nexus and CIF

The Nexus communication subsystem[12] of the Globus system[13] has recently been modified to provide a generic communication Application Programmer's Interface (API). This generic communication API has been developed to support some of the collaboratory development projects under way within the Department of Energy[1]. The generic communication API is now part of the Collaboratory Interoperability Framework (CIF) and provides a single uniform interface to a wide variety of communication protocols[2][3]. The CIF interface currently provides access to unicast and multicast protocols with several different levels of reliability. The reliable multicast protocols that have been included in the CIF environment to date are XTP and Totem. The unicast protocols include UDP, TCP, shared memory, and several supercomputer related communication mechanisms. The Nexus communication subsystem has been implemented in both C and Java and has been ported to most computer platforms.

2.1.5 Communication Services

Current communication protocols define generic services that provide either reliable or unreliable communication and multicast or unicast communication. These protocols are designed to serve the needs of a specific class of applications. Each protocol supports either unicast or multicast and either reliable or unreliable delivery. The protocol tries to support all applications that require the specific level of reliability and multiplicity that it provides. The delivery properties of a protocol are designed to serve the requirements of all the applications that might use the protocol. The only protocol that currently attempts to provide a spectrum of communication capabilities that span both unicast and multicast is the XTP protocol.

Since the sockets interface is not well suited to expressing multicast communication requirements, the application programming interface of each of the reliable multicast protocols is unique. Some protocols have attempted to maintain an interface similar to the sockets interface but extended to allow membership and multiple senders in a group. Other protocols have completely discarded the sockets interface and have defined an interface that is much more group oriented rather than point-to-point. The CIF interface provides an interface that falls somewhere between the two. It preserves the concept that communication takes place over a connection but it provides all the group semantics that are needed to work with multicast groups and unicast connections through a single consistent interface.

2.2 Visualization

Visualization is the process of transforming abstract data into meaningful images that promote scientific insight. Traditionally, the process of visualization spans several interrelated fields. Rendering is the process of transforming geometry, vectors and image-based data into an array of pixels. The applications that are being proposed for the NGI pose some formidable obstacles for visualization. The large datasets and distributed systems proposed will require the development of new forms of visualization and a bridging of the gap between the application and the underlying network layer. In the long term, visualization applications will have to be modified to use direct access to flexible communication protocols and network status information to adapt to network behavior. The application of new network aware techniques will also allow multiple sites with different network characteristics to view the data simultaneously, facilitating collaboration among researchers.

The size of the data sets in the NGI applications and the differing locations of the computational resources and the researchers preclude traditional forms of visualization where the data set is downloaded to the researcher site and rendered on the local hardware. In some cases, the data set will have to be rendered at one site and displayed remotely. The growth of these network based visualization systems introduces a wide variety of issues that have not been well addressed and investigated.

Most commercial visualization software packages such as AVS, IBM DX and Khoros assume that the user and the data are co-located. They typically provide a brute force mechanism for network based visualization, such as using X11 redirects to display at remote locations. They have no knowledge of the underlying networking infrastructure. Although this method works well over a high-bandwidth, low latency, and low loss network, it is not useable in most situations.. Latency and reliability issues leave the visualization too confusing and too frustrating for the end user, and the visualization system itself becomes a major stumbling block to the researchers' ability to understand the data. In addition, these types of systems rule out the possibility of collaborative visualization.

Network based visualization applications typically follow one of two approaches. In one approach, an application requests access to portions of data or geometry that will be transmitted across a network to be visualized on the local workstation. A second class of applications encompasses those that are capable of high-capacity data visualization, and that transmit images across the network to the user. There are hybrid forms of network based visualization applications that have combinations of the two approaches.

Current communication protocols provide generic services and do not take application specific requirements into account. Remote visualization is highly sensitive to issues such as latency and bandwidth and has a unique set of communication requirements. Latency affects several aspects of a remote visualization application. Navigation of three dimensional spaces and structures requires a fast feedback loop between the user and the rendering system. In a visualization system where the images are rendered remotely, there is latency associated with transmitting the images from the rendering system to the viewer. There is also a latency associated with user response time. When a user interacts with a system, the interaction must be sent over the network to rendering engine and then the results sent back to the user. Network latency and jitter has been shown to have a drastic effect on user performance in visualization environments[22]. High latency (>200msec) combined with jitter has the greatest impact on user performance. This is especially true in collaborative visualization environments.

In non-distributed visualization systems where the data and rendering engine are co-located, performance problems are most often caused by limitations in the rendering engine, computational limitations, memory transfer, disk access, etc. These effects are typically reproducible. Attempting to visualize the same dataset at different times will most often reproduce similar results in performance. The distributed nature of the NGI visualization applications however, brings variability and host of other causalities for variable performance[20]. The varying nature of both the networking and hardware infrastructure in NGI applications lends itself to the development of a visualization system that is flexible to both network and hardware variability. Unfortunately, most visualization systems do not currently support adaptation to varying conditions.

Rendering the data at a remote location and transmitting the images over a WAN for display at a remote location requires new techniques in adaptive visualization and in new communication protocols. A full screen resolution rendering consists of approximately 1280x1024 pixels, with each pixel being composed of at least 32bits, (alpha, red, blue

and green channels). With an update rate of at least 10 frames per second, a visualization system that simply transmits imagery without compression will consume at least 500Mbps of network bandwidth. If the user desires to see their visualization in stereographic form, the required bandwidth increases greatly since the refresh rate will now be at least 120Hz. The required bandwidth for the same 1280x1024 frame now mushrooms to 5Tbps. If the researcher desires to view their visualization in an immersive environment, like a CAVE with 4 display surfaces, the required bandwidth increases to 20Tbps. It becomes quickly obvious that a naïve approach to remote visualization does not scale very well.

In order to determine how to react to issues such as latency and jitter in a visualization system, we will need to quantify what is causing them and where they are occurring. This will require decomposing the different components of the entire system to determine where the problems, such as latency, jitter or bottlenecks are occurring. The development of end to end network monitoring systems that have been proposed in other NGI proposals potentially provides a visualization system the ability to monitor each step of the entire visualization process and make that information available to the user. Previous attempts to develop performance models for distributed visualization systems has modeled the performance of the system at the application layer[16]. The underlying networking layer remained a “black box”.

The “black box” approach to the networking layer potentially makes a network based visualization system unstable and unpredictable. Since the networking layer is completely hidden, the user must use other methods to determine points of failure and bottlenecks. Without reliable and predictable performance of an interactive application, the application quickly becomes perceived as unusable. The system itself becomes an obstacle to the task that it is attempting to accomplish, visualization and analysis of the data. A scientific researcher analyzing their data should not need to know how to track performance bottlenecks and network failures. In most cases, the scientific researcher would rather not use such a system, because it hinders the analysis of their data.

2.2.1 Network Aware Visualization Applications and APIs

Most visualization and virtual environment applications that have attempted to create a network aware system have been developed as one time systems. There has not been a toolkit that has attempted to make new communication protocols and techniques available to visualization applications.

2.2.2 CAVERNsoft

The closest system that attempts to address this issue is the CAVERNsoft[16] system. CAVERNsoft is a C++ hybrid-networking/database library optimized for the rapid construction of collaborative Virtual Reality applications. The CAVERNsoft system, however, does not provide mechanisms to modify and monitor the underlying network infrastructure. The CAVERNsoft system is also geared more toward creating virtual environments instead of visualization. The Visualization Toolkit we are proposing will be

developed such that it could be easily incorporated into the CAVERNsoft system for use in some of the NGI applications, thus enhancing the CAVERNsoft system and the NGI applications.

2.2.3 NPSNET and DIS

Most systems that attempt to have made efficient use of network resources have been simulation systems. The NPSNET system, developed by the Naval Postgraduate School, is used for simulation and training over a WAN[17]. NPSNET uses the Distributed Information System (DIS) protocol to maintain state among the participants. In the NPSNET system there are thousands of entities each with their own state information and agenda. The amount of network communication required to transmit state information to all of the participants in the system is staggering. NSPNET, however, was designed such that the virtual space was subdivided into smaller regions. These regions were mapped into different multicast groups. By knowing where a user was in the virtual space, the application could join the multicast groups that corresponded closest to its current location. As the user moved through the virtual space, the application joined and disconnected from different groups. This greatly reduced the communication costs between the participants.

2.2.4 GLR

Silicon Graphics has developed a network based visualization API called GLR[14] that renders imagery at one location and ships the imagery across a network to be displayed on the end user site. The API attempts to maximize the available bandwidth by rendering smaller thumbnails while the visualization is moving and rendering full resolution images when the visualization has stopped moving. This was done to reduce the latency in the user interaction. Unfortunately, GLR was developed for high bandwidth, low latency, jitter and loss networks. Its performance signficantly degrades when the used across a WAN. It also provides no feedback to the application about the state of the network. With the development of the Advanced Visualization Toolkit, it is quite possible that one could modify GLR to take advantge of the network aware aspects of the toolkit to increase performance of GLR over a WAN.

3 Preliminary Studies

LBNL has demonstrated network aware and distributed visualization systems over several high-speed network testbeds. The visualization application for the MAGIC project is a network aware terrain visualization system that has knowledge about the underlying network[15][27]. The application requests data from distributed servers located across a gigabit-speed WAN and renders the data at the user site. The rendering is done at the user site instead of the data site with the resulting images transmitted. This is done to avoid problems with network latency and jitter that can severely hamper user performance in interactive visualizations[18]. The MAGIC application achieves consistent rates of 30fps over a gigabit speed network. The major bottleneck in the system is not the available network bandwidth, as is typical in most distributed

visualization systems, but is instead the main memory into graphics memory transfer time. The application monitors the user movements and is able to pre-fetch the data from the distributed server system. A multi-resolution hierarchy was developed which allows for prioritization of the data being requested from the servers. This ensures that there is always some data that is available to be rendered. This application has been demonstrated on the MAGIC network, NTON, I-Way and the I-Grid.

LBNL has also experimented with collaborative and remote visualization applications. LBNL, in conjunction with Oak Ridge National Laboratory (ORNL) and ANL have demonstrated the capability to integrate audio and video conferencing along with semi-immersive applications over a WAN. Working with Lawrence Livermore National Laboratory (LLNL) and New York University (NYU), we have tested GLR running between LLNL and LBNL over the NTON (National Transparent Optical Network) and between LBNL and NYU via ESnet. Our trials measured the latency across the different networks and the affects that they had on an existing network visualization API.

4 Research Design and Methods

4.1 Communication Layer Toolkit

There are many projects currently under development that have the potential to significantly influence future communication protocol development. Most communication protocols currently work to hide most of the network behavior and complexity from the application. This helps to keep the application complexity down and keeps the knowledge required by the application programmer to very simple semantics. Unfortunately, this approach does not allow the application to obtain information regarding current network conditions making it difficult for the application to adjust to dynamically changing network conditions.

The simple interface approach assumes that the application is unable to handle the complexity introduced by exposure to network congestion, network bandwidth information, retransmission requests, etc. By building an application level toolkit that understands the application level needs we intend to interpret the complex network level information and use the information to adapt and effectively use the current networking capabilities of each remote user. The first step in building the application specific layer is building the underlying communication toolkit layer that provides the low-level network information and control to the application level toolkit. We will start with describing the components of the communication layer toolkit.

4.1.1 Performance

An essential criteria for the underlying communication toolkit layer is performance. We will provide an interface for requesting network capacity and performance measurements. The toolkit will make these measurements using the pathchar program of which there is already a public domain implementation called pchar (implemented at Sandia National

Lab). We will also provide an ability to obtain ongoing detailed performance data through an interface to the NetLogger utilities developed at Lawrence Berkeley Lab[24].

Each communication protocol included in the communication layer toolkit will be evaluated for performance on high bandwidth and low bandwidth connections. The multicast protocols will also be evaluated for performance when the group of receivers is located on a combination of high and low bandwidth connections. The evaluation criteria will be latency and throughput performance during transmission of test workloads that model the network load expected during a typical collaborative visualization. The model load will be determined by measuring the current load between the rendering engine and the display during a typical visualization. Since measuring the rendering engine to display load of a non-remote user will reflect the load of a non-networked visualization, this data should serve as a good worst case scenario for the network traffic loading for a remote collaborative visualization application.

4.1.2 Unicast Communication

Since UDP and TCP provide no support for priority communication we expect to incorporate XTP for the reliable and unreliable unicast communication. The decision of which protocol to incorporate will be determined based on their relative performance in tests. The advantage of using TCP and UDP is that these protocols are widely available and they are implemented in the kernel and fully debugged. It is possible that different platforms or conditions will favor one protocol over another. If this is the case, we will consider providing both TCP/UDP and XTP in the toolkit.

4.1.3 Reliable Multicast

The communication toolkit layer needs to provide a flexible reliable multicast service to the next higher layer to provide the performance and scalability desired. When dealing with reliable multicast in a high bandwidth delay product environment or in a moderately dynamic or lossy environment, a small change in the requested service guarantees can lead to a large gain in performance. For example, if the membership of a group is changing frequently and the delivery guarantee requested is for ordered message delivery with membership changes then the message delivery will be delayed each time there is a membership change. If the delivery guarantee is for reliable delivery at all members of the group and one member is behind the others in delivery then the buffering requirements of the protocol will grow and will force the protocol to eventually quit accepting new messages. If on the other hand the application was aware of these affects, it could take appropriate action to alleviate the situation. The application toolkit needs to be able to make informed choices regarding delivery service guarantees so that it can use the service that best fits its needs without sacrificing performance.

The InterGroup protocol[9] currently under development at UCSB as part of the CIF project is expected to provide reliable multicast capabilities to this project. The InterGroup protocol takes a somewhat radical departure from traditional reliable multicast protocol services. The InterGroup protocol allows each receiver in a multicast group to decide what level of reliability and ordering it would like to receive messages in.

This allows the different receivers to operate in modes appropriate to their needs and capabilities. A receiver that is experiencing moderate loss could drop down to unreliable delivery and quit trying to get all the messages reliably. This would allow the rest of the group to operate reliably without forcing the lossy receiver to leave the group. The InterGroup protocol also contains mechanisms that keep the membership down to a limited set by only keeping explicit membership for the group of processes that are sending messages. In the case of a visualization application this is likely to be the rendering engine or data source so the membership of the sending group is likely to be fairly static. InterGroup also contains mechanisms for allowing the user to specify the level of ordering desired: unordered, source ordered, or group ordered.

Currently the InterGroup protocol is being implemented in the Java programming language and so performance may become an issue. We intend to recode the critical tasks into C as needed to support the visualization applications. Also, InterGroup does not currently contain mechanisms that allow for prioritizing messages nor does it allow the next higher layer to use application level framing (ALF) to request retransmission of missing data while using a unreliable delivery service from the communication layer. We intend to add these features to the protocol as part of the development of the communication toolkit project. Since the protocol is already equipped to handle retransmission of all messages adding ALF capabilities will not be a significant change. The most significant change will be the interface that requests the message from other members of the group at the application toolkit layer when the communication layer no longer has the message in its buffers.

4.1.4 Security – Akenti

The Akenti protocol[23] currently under development at LBNL will be integrated into the communication layer toolkit to provide authorization of user capabilities. This will allow the application to define the group of participants that can participate in a particular communication session. For unicast connections the use of these mechanisms to keep unauthorized users from participating in the communication session will be relatively straight forward. Preventing unauthorized users from participating in the communication when the session is multicast is quite a bit harder. The Cliques project[8] is working on a promising mechanism for dynamic group key agreement and we are expecting to integrate the cliques protocol into the communication layer toolkit for multicast group communication security. The work to integrate the Cliques protocol and the Akenti protocol has been proposed under a separate basic technologies proposal.

4.1.5 Differentiated Services – integration

We do not intend to implement a differentiated services infrastructure in this project. We intend to integrate the differentiated services architecture as it becomes available in the network. As part of this work we intend to provide a communication toolkit layer interface that will allow the application toolkit layer to request a particular class of service and a bandwidth that it intends to use on that connection. The communication layer will take care of contacting the bandwidth broker for the local site and carrying on the negotiation. The application toolkit will be informed via a warning if the requested

bandwidth is not available as priority service. The application will also be provided feedback regarding what bandwidth is available for priority service to the desired destination so that the application can retry its reservation if desired. At the current time this differentiated service interface will only be available for unicast communication since differentiated services in a multicast environment is still an open research topic.

4.1.6 Congestion Control

The Internet is a heterogeneous environment and the bandwidth available between individual sites is dynamically changing. If a communication protocol does not continuously monitor the capabilities of the connection it is using it risks causing congestion on the connection. Once the connection becomes congested, the available throughput decreases. Retransmissions also require bandwidth and further decrease the bandwidth available for new data. Reliable delivery protocols must continuously monitor congestion to determine appropriate back-off parameters. The goal is to minimize the number of retransmissions needed and maximize the use of the bandwidth available. A reliable communication protocol also has to buffer messages until it knows they have been delivered. If the link is congested and has high loss then the buffering requirements increase and the delivery of messages can be delayed waiting for retransmitted messages.

We intend to incorporate into the communication layer protocols that have fair congestion control algorithms (where fair refers to its behavior with respect to other traffic). The significant change from current communication protocols is that the information regarding changes in congestion and flow control properties will be exposed to the next higher layer in the form of events. The next higher layer will be able to express a desired maximum, minimum and average traffic rate for each communication session. The application will be notified using event triggers when the protocol is unable to maintain the requested average traffic rate and again when it is unable to achieve the requested minimum traffic rate. In addition, the application will be supplied with an interface that will allow it to check the current estimated network throughput capabilities between itself and any other site and the current average throughput the protocol is actually achieving.

4.1.7 Common Interface Definition

A critical component of this project will be definition of the communication layer API. The communication layer toolkit will provide the infrastructure required for future application toolkits to be built. This requires that the communication layer, while providing all the functionality required by the visualization toolkit, must also provide a very general interface useable by other application specific toolkits. We will be building on our experience developing the Collaboratory Interoperability Framework (CIF) common communication interface and will start from that basic core[2]. But, the CIF interface was intended to provide a very simple interface whereas the communication layer we are building needs to expose the communication complexity to the application toolkit. The CIF interface will clearly need significant additions to meet our needs.

When establishing a communication session the visualization toolkit will add the ability

to specify the flow control, reliability, ordering, membership, priority, security requirements, and multiplicity desired for the session. The CIF API will be relatively simple to extend to add these parameters since definition of the communication parameters takes place before opening the communication session. The parameters are then passed to the communication session on open. Each of these parameters will have a default value defined so the application toolkit need only set the parameters it wants to be different from the default values.

Event services for notification of communication level events will also be added to the CIF API and the application toolkit layer will be provided an interface to register listeners for each type of event. The event classes will include flow control, membership, security, and retransmission. The API will include two classes of events, warnings and notifications. Events such as flow control violations or inability to achieve desired flow control parameters will fall under the class of warnings. Events such as membership changes and message arrival will fall under the class of notifications. By classing the events we will allow the application layer to define handlers at whatever level of granularity it prefers. It will be able to either register a handler by the general class of event or by the specific type of event.

4.2 Visualization

The obstacles associated with network based visualization are numerous and a complete implementation of an entire Toolkit that addresses all the problems is beyond the scope of this proposal. Instead, we will be working closely with the funded NGI application proposals to determine the types and forms of visualization that will be required.

The development of communication protocols that will create network aware visualization applications will require a layer that will enable visualization applications to be able to incorporate them into the applications. We will develop an API that will allow visualization applications access to the underlying networking communication protocols in a visualization context and will also enable the applications to have access to information about the underlying network infrastructure.

4.2.1 Decomposition Methods

The decomposition of the types of data in a distributed visualization application will be completed prior to the development of the API. This decomposition is essential to determine how to structure the underlying interfaces to the networking layer. We anticipate working closely with the funded NGI application proposals so that the types of data that we will be focusing on will be applicable to the applications. There are several basic types of data that we will be characterizing:

- User interaction
- User to user interaction (video/audio streams)
- Rendered images
- Geometry

It is possible that these basic types that we have defined will not be sufficient. We will work closely with the NGI application researchers to further refine the decomposition of the different types of data encountered in a distributed visualization system.

4.2.2 User Interactions

One of the application areas that we will be focusing on is in the transmission of user interaction information and of the generated imagery. In an interactive visualization, there needs to be a tight feedback loop between a user's interactions and the resultant changes on the display. Without a tight loop, there will be a delay between the time a user commits an action and the resultant change on the display. If the action and the resultant reaction are not closely coupled, the system will appear to the user to be unresponsive and will inhibit the visualization and analyis process.

The distributed nature of the NGI visualization applications introduces many opportunites for latency. The network is a potential source of latency, the size of the data set could inhibit the rate at which an image is rendered or retrieved from a storage system. Also, the network to memory transfer of the data could also introduce unacceptable latency. How these different components affect the overall latency of the system is an issue and how this information will be made available to the application and the user will be addressed in the Toolkit.

Using the Toolkit, the application will be able to prioritize some types of information that would be very suceptible to latency. For example, user motion in an immersive environment could be transmitted at a higher priority since a high latency response to some action could translate into disorientation and physical discomfort to the viewer. Also an application could send a request that any request for repairs of missing data be dropped because it is no longer valid in the current context.

4.2.3 Image, Video and Audio Streams

The distributed nature of the NGI visualization applications requires means of collaboration between researchers in different location. Since the audio and video tools for collaboration will most likely be simultaneously competing for resources with the visualization application on the same platform, coordination between the conferencing and visualization applications would most efficiently utilize the limited network and hardware resources. One method of coordination would be the prioritization of the information between the video, audio and data streams. The user could decide that the video stream should take precedence over the visualization and vice versa.

If the system is a remote rendering system, where the visualization is rendered on one platform and then the rendered images are sent across a WAN to be displayed on a display device at the end user site, there could be different encodings of the images. During an interactive application, there is always a tradeoff between reliability and interactive rates. Reliability typically requires request for repairs and re-transmission of data. This delay lowers the interactive rate of the system.

Most current distributed visualizations systems do not allow the application nor the user the ability to decide how this trade off should occur. The toolkit will allow to user to sacrifice interactivity for reliability, where every single frame is guaranteed to be shown. Conversely, the toolkit will also allow the user to sacrifice reliability for interactivity. The major difference in these scenarios compared to current systems, is that the end user has control over how the application behaves. The application will no longer be as unpredictable and beyond the user's control. We believe that this will greatly enhance the usability of network based visualization systems and their acceptance by the scientific research community.

4.2.4 Geometry

Distributed visualization systems are highly susceptible to latency and jitter. One way that some distributed visualization systems attempt to avoid latency issues is by rendering the visualization at the end user location. Geometry or graphics calls are sent over a WAN to the end user location and are rendered on the hardware at the end user site. Since the visualization is occurring at the end user site, the loop between user interaction to seeing the results of the interaction is considerably shorter. Typical methods in these types of systems is to transmit lower resolution geometry first and then, when time permits, to transfer the high resolution geometry. This guarantees that the user will be able to see and manipulate the visualization, albeit at a low resolution. As soon as the viewer stops moving, the high resolution gets ships across the WAN and upon its arrival is rendered on the screen. By using the toolkit, the application could determine which method would be the most efficient method, to render locally or render remotely.

4.2.5 Application Programming Interface

We will be investigating how different forms of user information can be priortiized and how the underlying communication protocols can assist in prioritizing the data. The application will need methods to interact with the underlying communication protocols to prioritize the data. We will be providing an API that will allow the applications to decided what type of information will need to be prioritized.

In order to be able to determine what type of data should be prioritized, the application will need to know how the underlying network is performing. The Toolkit API will provide integration points for network monitoring tools such like that are being developed in other NGI proposals. The NetLogger network monitoring tool being proposed by LBNL and KU for the NGI is a prime example of how an application can receive information about the underlying network state. The Toolkit will provide entry points for the application to be able to interface to such a tool if it is integrated into the Toolkit.

4.2.6 Using the Communication Infrastructure

How visualization applications could most optimally utilize the communication infrastructure will be investigated in this proposal. We will be testing different methods of utilizing the communication infrastructure in the context of visualization. This testing will effect how we decompose the visualization data and the development of the API to

encapsulate the communication infrastructure. We will investigate the type of information required by the visualization application about the network status and how to present this information to the application.

We will be investigating the most effective ways of notifying the application about changes in the network status. Since visualization applications are real time applications and typically stress the hardware systems, we do not want to be constantly interrupting the visualization system with messages about the current network status. This overhead could cause more harm then good. There are times, however, when the application should immediately know if there is a critical problem, such as loss of connectivity to one of the distributed resources. We will provide hooks through the API to allow the application to decide how it will want to handle the information that will be provided to it by the underlying layers and distributed resources. Critical situations could be immediately brought to the attention of the user.

5 Summary

The applications that are being proposed for NGI type networks will require seamless collaborative access to data and will require the ability to visualize data remotely where the data must travel across a high-speed wide-area network to reach the user. The inherent networking component of these applications introduces a new set of problems that have not been adequately addressed in the past. These new distributed visualization applications will require a closer coupling between the visualization and the underlying networking infrastrucutre.

The data visualization communication toolkit will allow visualization applications to have direct access to network status information and to directly control communication protocol and network behavior. Visualization applications will be able to adapt to a changing networking infrastructure and provide real-time feedback to the user regarding the state of the network. The toolkit will also provide options for the visualization software to modify its behavior to reflect the current networking conditions. These capabilities will also allow there to be multiple sites with different network characteristics collaborating and viewing the data simultaneously.

We believe the data visualization communication toolkit is the first step in addressing the issues regarding distributed visualization over a wide-area network. The toolkit will be developed so that it can be incorporated into the NGI applications and its functionality will be extensible to allow integration of new networking technologies as they are developed.

6 Potential Collaborations (Collaboration Arrangements)

The level of effort required to design and build projection versions of the complete Advanced Visualization Toolkit is beyond funding currently available to a single proposal. The research proposed here complements other proposals submitted by colleagues at ANL, LBNL, ISI, and other institutions. These proposals have been

developed with the collective goal of defining and implementing an *Integrated Grid Architecture* for advanced network applications. This architecture promotes the development of high-performance, reliable, network-aware application and the sharing of code across disciplines by the definition of a layered architecture comprising four principal components:

- At the *Grid Fabric* level, primitive mechanisms provide support for high-speed network I/O, differentiated services, instrumentation, etc.
- At the *Grid Services* level, a suite of Grid-aware services implement basic mechanisms such as authentication, authorization, resource location, resource allocation, and event services.
- At the *Application Toolkit* level, toolkits provide more specialized services for various application classes: e.g., data-intensive, remote visualization, distributed computing, collaboration, problem solving environments.
- Finally, specific *grid-aware applications* are implemented in terms of various Grid Services and Application Toolkit components.

Our experience developing and using both successful Grid services (e.g., Globus) and substantial Grid applications convinces us that the definition of such an Integrated Grid Architecture is essential if the scientific community is to adopt and profit from NGI environments. Without this architecture, we will continue to see a range of inadequate, fragile, stovepipe systems. With it, we can hope to see broad deployment and adoption of fundamental basic services such as security and network quality of service, and sharing of code across different applications with common requirements. We believe that the DOE's NGI program represents an unprecedented opportunity to create and deploy such an Architecture, and have developed our proposals with this goal in mind.

The Advanced Visualization Toolkit will be complementary to much of the work being proposed in complementary proposals. The Toolkit could be integrated into the following proposed application proposals, assuming they are funded beyond a single year:

- Prototyping a Combustion Corridor (led by LBNL)
- CorridorOne: An Integrated Distance Visualization Environment for SSI and ASCI Applications (led by ANL)
- The Earth System Grid (led by NCAR)

The following basic research NGI proposals could also be integrated into the Advanced Visualization Toolkit so that their capabilities could be made available to distributed visualization applications. This integration effort would be beyond the scope of this project and would be potentially be completed in the application proposals.

- Network Monitoring for Performance Analysis and for Enabling Network-Aware Applications (LBNL/KU)
- A Bandwidth Reservation System (LBNL)
- A Uniform Instrumentation, Event, and Adaptation Framework for Network-Aware Middleware and Advance Network Applications (ANL/UIUC)

- Diplomat: Policy-Based Resource Management for Next-Generation Internet Applications (ANL/ISI/Wisconsin)

7 Literature Cited

[1] D. Agarwal. "Collaborating Across the Miles." *Proceedings of the INMM/ESARDA – Workshop on Science and Modern Technology for Safeguards.* Albuquerque, NM, September 1998.

[2] D. Agarwal, I. Foster and T. Strayer. "Standards-Based Software Infrastructure for Collaborative Environment and Distributed Computing Applications." White paper. http://www-itg.lbl.gov/~deba/publications/OOSB.white.paper.html.

[3] D. Agarwal, K. Berket, N. Narasimhan, P. Schabert, I. Foster, and S. Tuecke. "The Collaboratory Interoperability Framework Common Application Programming Interface." http://www-itg.lbl.gov/CIF/Reports/GcommonAPI.html

[4] D. A. Agarwal, P. M. Melliar-Smith, L. E. Moser, and R. Budhia, "Reliable Ordered Delivery Across Interconnected Local-Area Networks," Transactions on Computer Systems, vol. 16, no. 2 (May 1998).

[5] D. Agarwal, S. Sachs, and W. Johnston. "The Reality of Collaboratories." *Computer Physics Communications*, Vol. 110, Issue 1-3. May 1998, pp 134-141.

[6] Y. Amir, L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal and P. Ciarfella, " The Totem Single-Ring Ordering and Membership Protocol," ACM Transactions on Computer Systems 13, 4 (November 1995), 311-342.

[7] Y. Amir and J. Stanton, "The Spread Wide Area Group Communication System," Technical Report CNDS-98-4 available from ftp://ftp.cnds.jhu.edu/pub/papers/spread.ps.

[8] G. Ateniese, M. Steiner and G. Tsudik, "Authenticated Group Key Agreement and Related Issues," ACM CCCS'98, San Francisco, November 1998.

[9] K. Berket, L. E. Moser and P. M. Melliar-Smith, "The InterGroup Protocols: Scalable Group Communication for the Internet." *Proceedings of the Third Global Internet Mini-Conference* (in conjunction with Globecom '98), Sydney, Australia, November 8-12, 1998

[10] D. Dolev, D. Malki, "The Design of the Transis System," Proceedings of Dagstuhl Workshop on Unifying Theory and Practice in Distributed Computing, September 1995.

[11] Floyd, S., Jacobson, V., Liu, C., McCanne, S., and Zhang, L., "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing," IEEE/ACM Transactions on Networking, December 1997, Volume 5, Number 6, pp. 784-803.

[12] I.Foster, C. Kesselman, S. Tuecke, " The Nexus Task-Parallel Runtime System," Proceedings of the 1st Int'l Workshop on Parallel Processing, pg. 457-462, 1994.

[13] I. Foster, C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit," International Journal of Supercomputer Applications, 11(2):115-128, 1997.

[14] M. Kilgard, "GLR, an OpenGL Render Server Facilitiy", Silicon Graphics, (http://reality.sgi.com/opengl/glr/glr.html)

[15] Y. Leclerc, S. Lau, "TerraVision: A Terrain Visualization System", AIC Technical Note 540, SRI International, Menlo Park, CA, 1994.

[16] J. Leigh, A. E. Johnson, T.A. DeFanti, “Issues in the Design of a Flexible Distributed Architecture for Supporting Persistence and Interoperability in Collaborative Virtual Environments”, *Proceedings of Supercomputing '97,* Nov 15-21, 1997, San Jose, California.

[17] M. Macedonia, M. Zyda, D. Pratt, D. Brutzman and P. Barham, "Exploiting Reality with Multicast Groups," *IEEE Computer Graphics Applications*, September 1995, pp.38-45.

[18] M. Macedonia and M. Zyda, "A Taxonomy for Networked Virtual Environments," *IEEE Multimedia*, Volume 4, No. 1, January - March 1997, pp. 48-56.

[19] K. Miller, K. Robertson, A. Tweedly, and M. White, “StarBurst Multicast File Transfer Protocol (MFTP) Specification--An Internet Draft,” available from http://www.ietf.org/internet-drafts/draft-miller-mftp-spec-03.txt. Information on the Starburst protocol is also available at http://www.starburstcom.com/.

[20] K. Park, K. and R. Kenyon, “Effects of Network Characteristics on Human Performance in a Collaborative Virtual Environment”, Proceedings *of IEEE VR '99*, Houston TX, March 13-17, 1999.

[21] W.T. Strayer, S. Gray, and R.E. Cline, Jr., “An Object-Oriented Implementation of the Xpress Tranfer Protocol,” Proceedings of the Second International Workshop on Advanced Communications and Applications for High-Speed Networks (IWACA), Heidelberg, Germany, September 26-28, 1994.

[22] V. E. Taylor, R. Stevens, and T. Canfield, "Performance Models of Interactive, Immersive Visualization for Scientific Applications," *Proceedings High Performance for Computer Graphics and Visualization Conference*, July 3-4, 1995, Swansea, U.K

[23] M. Thompson, W. Johnston, S. Mudumbai, G. Hoo, and K. Jackson, “Certificate-based Access Control for Widely Distributed Resources,” submitted to the Usenix Security Symposium '99. Mar. 16, `99. (http://www-itg.lbl.gov/security/Akenti/docs/UsenixSec99.pdf)

[24] B. Tierney, W. Johnston, B. Crowley, G. Hoo, C. Brooks, and D. Gunther, “The NetLogger Methodology for High Performance Distributed Systems Performance Analysis,” Published in the Proceedings of the IEEE HPDC-7’98, July 1998, Chicago, Illinois, 28-31.

[25] K. Watsen and M. Zyda, "Bamboo - A Portable System for Dynamically Extensible, Networked, Real-Time, Virtual Environments," *Proceedings of VRAIS 98*, 16 - 19 March 1998, Atlanta, GA, pp. 252-259.

[26] B. Whetten, T. Montgomery, and S. Kaplan, “A High Performance Totally Ordered Multicast Protocol,” Dagstuhl Castle, Germany, pp. 33-57. Springer Verlag, Lecture Notes in Computer Science 938, 1994. Current protocol information available at http://www.gcast.com/.

[27] The Magic Network, (http://www.magic.net)

[28] “The Multicast Dissemination Protocol (MDP) Framework,” MDP protocol information available at http://manimac.itd.nrl.navy.mil/MDP/.

8 Glossary

ACTS	Advanced Computational Testing and Simulation
AMRVIS	a visualization and data analysis tool for examining data files generated by the Adaptive Mesh Refinement (AMR) code.
ANL	Argonne National Laboratory
API	Application Program Interface
ATM	Asynchronous Tranfer Mode
BAGNET	Bay Area Gigabit Network
BNL	Brookhaven National Laboratory
CAVERNsoft	CAVERNsoft is most succinctly described as a C++ hybrid-networking/database library optimized for the rapid construction of collaborative Virtual Reality applications.
CCSE	Center for Computational Sciences and Engineering, LBNL
CM	Cache Manager
CO_2	Carbon Dioxide
CSMI	Combustion Simulation and Modeling Initiative
DARPA	Defense Advanced Research Projects Agency
DiffServ	Differentiated Service
DOE	U.S. Department of Energy
DPSS	Distributed-Parallel Storage System
DVC	Data and Visualization Corridors
EETM	End-to-End Storage Manager
EMERGE	Esnet/MREN Regional Grid Experimental NGI Testbed
ESnet	Energy Sciences Network
GAA	Generic Authorization and Access
GARA	Globus Architecture for Reservation and Allocation
GASS	Global Access to Secondary Storage
GRAMs	Globus Resource Allocation Managers
GUSTO	Globus Ubiquitous Supercomputing Testbed Organization
HENP	High Energy Nuclear Physics
HENP-GC	High Energy Nuclear Physics – Grand Challenge
HPSS	High Performance Storage System
I/O	Input/Output
ICAIR	International Center for Advanced Internet Research, Northwestern University
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPG	Information Power Grid
I-WAY	Information Wide Area Year, event at Supercomputing ‘95
LAN	Local Area Networks
LBNL	Lawrence Berkeley National Laboratory
MAGIC	Multidimensional Applications and Gigabit Internetwork Consortium
MDS	Metacomputing Directory Service
MPLS	Multiprotocol Label Switching
MREN	Metropolitan Research and Education Network

MSS	Mass Storage Systems
NASA	National Aeronautics and Space Administration
NCAR	National Center for Atmospheric Research
NERSC	National Energy Research Scientific Computing Center
NGI	Next Generation Internet
NOx	Nitrogen Oxide
NREN	NASA Research and Education Network
NSF	National Science Foundation
NSP	Network Service Provider
NTON	National Transparent Optical Network
OC-12	An OC-12 circuit (622,000,000 bits per second) is bandwidth that was experimented with in the Gigabit Testbeds of the early 1990s. At the beginning of 1998 it is also the bandwidth of Sprintlink and MCI's backbones. By the end of 98, it should equal the speed of every major NSP's backbone.
OC-3	An OC-3 circuit (155,000,000 bits per second) is the backbone speed that major NSPs have upgraded their backbones to by the end of 1997.
OC-48	An OC-48 circuit (2,400,000,000 bits or 2.4 gigabits per-second) is the typical speed for many aggregated telephone voice circuits on inter city fiber optic lines. Before the end of the decade most NSPs should be operating at OC-48 speeds. A few are expected to implement OC-48 before the end of 1998.
OPTIMASS	Enabling technology for rapid analysis of earth science and environmental data
OSI	Open Systems Interconnection
PACI	Partnerships for Advanced Computational Infrastructure
PHENIX	PHENIX is one of four RHIC detectors under construction at Brookhaven National Laboratory (BNL) in Upton, New York
PI	Principal Investigator
QE	Query Estimator
QM	Query Monitor
QoS	Quality of Service
RSVP	Reservation Protocol
SLA	Service Level Agreement
SM	Storage Manager
SNL	Sandia National Laboratory
SSI	Strategic Simulation Initiative
STACS	Storage Access Coordination System
STAR	Solenoidal Tracker At Rhic
STAR-TAP	Science, Technology And Research Transit Access Point
TCP	Transport Control Protocol
UCAID	University Corporation for Advanced Internet Development
UIC/EVL	University of Illinois at Chicago, Electronic Visualization Laboratory
UWM	University of Wisconsin at Madison
WAN	Wide-area networks

9 Budget and Budget Explanation

Lawrence Berkeley National Laboratory

The three-year budget includes partial funding for one of the Principal Investigators to analyze the decomposition of visualization data and the development of the Visualization Toolkit API. It also funds working with the funded proposal applications to determine their visualization needs. It also includes partial funding for the other principal investigator and another staff to develop the communication layer toolkit that will provide the infrastructure for the Visualization Toolkit.

Travel funding for two trips per year to deployment sites for consultation, two trips per year to collaborating institutions, and one trip per year to a professional conference is included.

10 Milestones

The proposed work is a three year effort. An emphasis will be placed on early development and deployment of prototypes so that system usage can be understood and improvements made in response to feedback from application developers.

Year 1 Milestones

- Define prototype communication layer interface API
- Develop a workload to be used in evaluating communication protocols
- Evaluate performance and interfaces of existing communication protocols
- Develop a prototype version of the communication toolkit with limited functionality
- Identify the visualization requirements of the funded NGI applications
- Work with funded application researchers to determine their visualization needs
- Identify how a visualization can take advantage of network information
- Determine data decomposition
- Develop a prototype API that addresses the visualization requirements
- Make the prototype API available for incorporation into the NGI visualization applications
- Package and release V1.0 of the Advanced Visualization Toolkit API

Year 2 Milestones

- Implement the components of the prototype API.
- Implement further components of the communication toolkit
- Continue to work with funded application proposals to develop the Visualization Toolkit further
- Incorporate Visualization Toolkit into the CAVERNsoft system
- Demonstrate the ability for a visualization application to take advantage of the underlying networking infrastructure.
- Make the implemented API available for incorporation into the NGI visualization applications
- Package and release V2.0 of the Advanced Visualization Toolkit API

Year 3 Milestones

- Continue to work with NGI funded application researchers to determine the changing visualization needs
- Continue to implement portions of the API
- Develop links to network monitoring packages to allow visualizations to monitor the network status
- Package and release V3.0 of the Advanced Visualization Toolkit API

11 Other Support of Investigators

Dr. Deborah Agarwal is currently funded on two projects. She is funded to work part time on the Collaboratory Interoperability Framework Project and part time on the Distributed Collaboratories Project. These projects are on-going projects funded by the Office of Energy Research, Office of Computation and Technology Research, Mathematical, Information, and Computational Sciences Division, of the U. S. Department of Energy under contract No. DE-AC03-76SF00098 with the University of California.

Stephen Lau is currently funded by the Director, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy's Office of Computational and Technology Research and its Mathematical, Information, and Computational Sciences Division of the U. S. Dept of Energy under contract No. DE-AC03-76SF00098 to support and assist NERSC users in the area of visualization.

12 Biographical Sketches

Stephen Lau, Jr.

National Energy Research Scientific Computing Research Division +1 510 486-7178
Lawrence Berkeley National Laboratory +1 510-486-6363 fax
One Cyclotron Road, MS: 50F slau@lbl.gov
Berkeley, CA 94720 http://www-vis.lbl.gov/~slau

Education

- **B.A. Psychology (Cognitive Science), University of California, San Diego, 1989**

Current Position

- **Computer Systems Engineer III, Visualization Group, NERSC, 1996-Present**

Research Interests

Distributed visualization, network aware visualization applications, immersive environments, user interfaces, large scale distributed data visualization systems, cognitive aspects of user interactions to virtual environments.

Narrative

Stephen Lau, Jr. is a member of the Visualization Group at the National Energy Scientific Research Computing Center at Lawrence Berkeley National Labs where he investigates issues related to wide area network visualization and collaboration. Prior to LBNL, he was at SRI International where he worked on the DARPA MAGIC project to develop a wide area network based terrain visualization system of large data sets called TerraVision. TerraVision was a distributed terrain visualization system that accessed gigabyte datasets across a gigabit speed WAN. It achieved interactive rates by pre-fetching information from the servers and rendering the imagery locally. He also developed multicast visualization tools that allowed researchers to collaborate over a WAN and multi-user shared virtual environments.

Selected Professional Activities

- Network Chair (1996-Present) IEEE Visualization Conference
- Member of ACM

Related Publications and Presentations

- “The Use of Rear Projected Visualization Systems at Lawrence Berkeley National Laboratory for Research and Scientific Visualization”, S. Lau, E.W. Bethel, N. Johnston, T. Ligocki, D. Robertson, International Projective Technology Workshop, 1999, LBNL-42938
- "Parallelization of Radiance for Real Time Walkthroughs of Lighting Visualizations", S. Lau, D. Robertson, K. Campball, et. al, Poster Session, Supercomputing 1998
- “TerraVision on the I-Grid”, Demonstration at Supercomputing 1998, Orlando, FL.
- “An Overview of Issues Related to Remote Visualization”, presented at Dept of Energy Comptuer Graphics Forum, St. Michaels, MD, 1998.
- “VRML for Wide Area Scientific Visualization”, presented at Dept of Energy Computer Graphics Forum, Jackson Hole, WY, 1997.
- “Creating a Korea-U.S. International Testbed for Teleseminars and Collaboration”, B. Denny, S. Lau, N. Plotkin, et al, Korean International High Speed Networking, 1996.
- “TerraVision on the I-Way”, Demonstration at Supercomputing 1995, San Diego, CA.
- “TerraVision: A Terrain Visualization System", Y. Leclerc, S. Lau, AIC Technical Note 540, SRI International, Menlo Park, CA, 1994.
- “The MAGIC Project”, ARPA HPC Networking ’93.

Dr. Deborah A. Agarwal
Lawrence Berkeley National Laboratory
MS50B-2239 +1 510 486 7078
One Cyclotron Road +1 510 486 6363 fax
Berkeley, CA 94720 DAAgarwal@lbl.gov

Education

Ph.D. Electrical and Computer Engineering (Communication Networks)
University of California, Santa Barbara
August 1994
Advisor: Professor Louise E. Moser
Dissertation Title: Totem: A Reliable Ordered Delivery Protocol for Interconnected Local-Area Networks

M.S. Electrical and Computer Engineering
University of California, Santa Barbara, *April 1991*

B.S. Mechanical Engineering
Purdue University, *May 1985*

Research Experience

Staff Scientist

THE LAWRENCE BERKELEY NATIONAL LABORATORY
October 1994 - Present
Imaging and Distributed Collaboration Group.
Project leader responsible for design and development of a "Collaboratory Interoperability Framework." This project is a multilab effort and it focuses on developing a common service infrastructure for the collaboratory environment. My team is providing the common communication library including reliable and unreliable delivery of unicast and multicast messages. (http://www-itg.lbl.gov/CIF)

Project leader responsible for design and development of software to allow remote experimentation and collaboration between sites connected by a wide-area network. This project is creating a virtual laboratory in which users no longer need to travel to conduct experiments. The remote operators have monitoring and control apabilities along with an Internet multicast videoconferencing link between sites. (http://www-itg.lbl.gov/Collaboratories)

Developing WWW pages to aid people learning how to use the Internet Multicast Backbone and the videoconferencing tools vic, vat, and sdr. (http://www-itg.lbl.gov/mbone)

THE LAWRENCE BERKELEY NATIONAL LABORATORY
July 1993 - August 1993
Computer Networking Group.
Designed and developed an analysis tool for Internet multicast routing. The ability to multicast is new to the Internet and is still experimental. The debugging tool aids in network configuration and diagnosis of problems.

Communication Protocols, UCSB

Summer 1991 - Fall 1994
Developed a communication protocol for fault-tolerant wide-area networks. This protocol is used in asynchronous distributed systems that require reliable ordered delivery of messages. An implementation of the protocol was completed. Topics of interest include protocol services, routing, flow control, failure recovery and membership.

Publications

- "Totem: A Reliable Ordered Delivery Protocol for Interconnected Local-Area Networks," Ph.D. dissertation, University of California, Santa Barbara, December 1994.
- "The Totem Multiple-Ring Ordering and Topology Maintenance Protocol," with P. M. Melliar-Smith, L. E. Moser, and R. Budhia, Transactions on Computer Systems, vol. 16, no. 2 (May 1998).
- "The Reality of Collaboratories," with S. R. Sachs, and W. E. Johnston, Computer Physics Communications vol. 110, issue 1-3 (coverdate May 1998), pages 134-141.
- "Totem: A Fault-Tolerant Multicast Group Communication System," with L. E. Moser, P. M. Melliar-Smith, R. K. Budhia, and C. A. Lingley-Papadopoulos, Communications of the ACM, April 1996.
- "The Totem Single-Ring Ordering and Membership Protocol," with Y. Amir, L. E. Moser, P. M. Melliar-Smith, and P. Ciarfella, ACM Transactions on Computer Systems 13, 4 (November 1995), 311-342.
- " Reliable Ordered Delivery Across Interconnected Local-Area Networks," with L. E. Moser, P. M. Melliar-Smith, and R. Budhia, Proceedings of the International Conference on Network Protocols, Tokyo, Japan (November 1995), 365-374.
- "Extending Virtual Synchrony," with L. E. Moser, Y. Amir and P. M. Melliar-Smith, Proceedings of the 14th IEEE International Conference on Distributed Computing Systems, Poznan, Poland (June 1994), 56-65.
- "Debugging Internet Multicast," with S. Floyd, Proceedings of the 22nd ACM Computer Science Conference, [-0.5mm] Phoenix, AZ (March 1994), 22-29.
- "Fast Message Ordering and Membership Using a Logical Token-Passing Ring," with Y. Amir, L. E. Moser, P. M. Melliar-Smith and P. Ciarfella, Proceedings of the 13th International Conference on Distributed Computing Systems, Pittsburgh, PA (May 1993), 551-560.

13 Description of Facilities and Resources

The Advanced Visualization Toolkit will use existing infrastructure at participating NGI sites. We anticipate leveraging off of existing networking testbeds and funded NGI infrastructure proposals. The existing infrastructure at the different collaboration sites are listed below:

Lawrence Berkeley National Laboratory (LBNL)

LBNL computing facilities include the computational resources of the National Energy Research Scientific Computer Center (NERSC). This includes a massively parallel Cray T3E-900, with 640 application processing elements, each capable of performing 900 MFlops., a cluster of six Cray J90 machines that have a total of 160 vector processors, an 8 processor high performance rendering engine, a 106 TB capacity tape archive, multiple TBs of disk cache. LBNL also has available an ImmersaDesk and a rear projection Wall for display and interaction with semi-immersive visualizations. LBNL and ANL are also connected via an ESnet OC-12. LBNL also has connectivity to NTON. LBNL has 2 large Linux PC clusters, the PDSF with 48 nodes, and a PCP cluster, with 36 nodes. There is also a 300 gigabyte, four server DPSS cache system which is directly connected to NTON and ESnet and can provide over 450 Mbits/second of cache storage bandwidth.

Argonne National Laboratory (ANL)

Relevant Argonne computing facilities include a 128-node SGI Origin 2000 and 150-node IBM SP (for a total of around 100 Gigaops); 80 TB-capacity tape archive; multiple TBs of disk (of which we expect to configure at least 1 TB for use as a disk cache for this project); multiple multiprocessor Sun systems, which will be available for networking research purposes; and various production and experimental networking equipment that will support the networking research proposed for this project. The Argonne OC-12 ESnet connection, multiprocessor Sun E4000 system, and networking equipment were used to support the recent 320 Mb/s transfer rate ANL-LBNL networking experiments. ANL also has a 4-wall CAVE immersive system as well as an ImmersaDesk semi-immersive system and numerous graphics workstations.

Sandia National Laboratory (SNL-CA)

We will be building upon the existing combustion researcher workstation infrastructure at Sandia. The Lab is connected to ESnet via OC3, which will most likely be upgraded to OC12 sometime during FY00. We are also anticipating that LBNL and Sandia will be connected by an OC48 NTON connection. ESNet will be leading the effort for these connections. The required hardware to extend the networking testbeds is being proposed in a complementary NGI proposal. SNL-CA also has a large screen visualization projection system known as a Visionarium.